

AFRL-VA-WP-TP-2002-334

**MODEL PREDICTIVE CONTROL FOR
DYNAMIC UNRELIABLE
RESOURCE ALLOCATION**

David A. Castañón
Jerry M. Wohletz



DECEMBER 2002

Approved for public release; distribution is unlimited.

© 2002 IEEE

This work is copyrighted. The United States has for itself and others acting on its behalf an unlimited, paid-up, nonexclusive, irrevocable worldwide license. Any other form of use is subject to copyright restrictions.

20030128 193

**AIR VEHICLES DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542**

REPORT DOCUMENTATION PAGE				<i>Form Approved</i> OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YY) December 2002		2. REPORT TYPE Conference Paper Preprint		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE MODEL PREDICTIVE CONTROL FOR DYNAMIC UNRELIABLE RESOURCE ALLOCATION				5a. CONTRACT NUMBER F33615-01-C-3149	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER 69199F	
6. AUTHOR(S) David A. Castañón Jerry M. Wohletz				5d. PROJECT NUMBER ARPF	
				5e. TASK NUMBER 04	
				5f. WORK UNIT NUMBER T1	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Alphatech, Inc. 50 Mall Road Burlington, MA 01803				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Vehicles Directorate Air Force Research Laboratory Air Force Materiel Command Wright-Patterson Air Force Base, OH 45433-7542				10. SPONSORING/MONITORING AGENCY ACRONYM(S) AFRL/VACA	
				11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S) AFRL-VA-WP-TP-2002-334	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES Conference paper proceedings of the IEEE Conference on Decision and Control, Las Vegas, NV, December 10, 2002. © 2002 IEEE. This work is copyrighted. The United States has for itself and others acting on its behalf an unlimited, paid-up, nonexclusive, irrevocable worldwide license. Any other form of use is subject to copyright restrictions.					
14. ABSTRACT In this paper, we consider a class of unreliable resource allocation problems where resources assigned may fail to complete a task, and the outcomes of past resource allocations are observed before new resource allocations are selected. The resulting temporal allocation problem is a stochastic control problem, with a state space and control space that grow exponentially in cardinality with the number of tasks. We introduce an approximation by enlarging the admissible control space, and show that this approximation can be solved exactly and efficiently. The approximation is used in a model predictive control (MPC) algorithm. For single resource problems, the MPC algorithm completes over 98 percent of the task value completed by an optimal dynamic programming algorithm in over 1,000 randomly generated problems. On average, it achieves 99.5 percent of the optimal performance while requiring over 6 orders of magnitude less computation.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT: SAR	18. NUMBER OF PAGES 12	19a. NAME OF RESPONSIBLE PERSON (Monitor) Mark Mears 19b. TELEPHONE NUMBER (Include Area Code) (937) 255-8685
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			

Model Predictive Control for Dynamic Unreliable Resource Allocation¹

David A. Castañón² and Jerry M. Wohletz³

Abstract

In this paper, we consider a class of unreliable resource allocation problems where resources assigned may fail to complete a task, and the outcomes of past resource allocations are observed before new resource allocations are selected. The resulting temporal allocation problem is a stochastic control problem, with a state space and control space that grow exponentially in cardinality with the number of tasks. We introduce an approximation by enlarging the admissible control space, and show that this approximation can be solved exactly and efficiently. The approximation is used in a model predictive control (MPC) algorithm. For single resource problems, the MPC algorithm completes over 98% of the task value completed by an optimal dynamic programming algorithm in over 1000 randomly generated problems. On average, it achieves 99.5% of the optimal performance while requiring over 6 orders of magnitude less computation.

1 Introduction

A common assumption in resource allocation problems such as multiprocessor scheduling or job shop scheduling is that, once a resource works on a task, the task will be completed successfully. However, there are many resource allocation problems where resources can fail to complete the task, and further resource assignments are required for that task. We refer to this class of problems as *unreliable resource allocation problems*. Examples of these problems include assignment of search activity (e.g. sonobuoys) to sectors [12], assignment of ground-air missiles to aircraft, or more general assignment of weapons to targets [5] in diverse military applications. In this paper, we are interested in the problem of *dynamic* resource allocation where resources are non-renewable and unreliable, and where the success of past resource allocations can be observed before new allocation decisions are made. These problems re-

quire selecting which tasks to process first, and what resources to hold in reserve for allocation after observing the success of early allocations.

There is an extensive literature on many variations of weapon target assignment problems that are formulated as unreliable resource allocation problems. However, most of these variations consist of static problems [5] where no information on allocation outcomes is observed. Dynamic variations of these problems where outcomes of allocations were observed were studied in [6] and [2]. Recently, Murphey [8, 9] has addressed stochastic dynamic weapon assignment problems where new tasks arrive over time, and weapon assignments are unreliable. Murphey allows for observation of the new task arrivals, but no observation of the past allocation outcomes. The resulting problem formulation is a stochastic program [10]. The search theory literature has extensive results on dynamic search problems; a good overview of these results is available in [12]. However, the results focus on dynamic search focus on sequential search for a single object, where search resources are allocated to a single site at a time.

We focus on allocating unreliable resources to multiple tasks over two stages, where the outcomes of resources assigned in the first stage are observed before the second stage allocations are selected. We pose the problem as a stochastic control problem; although this problem can be solved using stochastic dynamic programming (SDP) [1], the number of states grows exponentially with the number of tasks. We propose an alternative algorithm, based on an approximate SDP formulation that can be solved in complexity nearly linear with the number of tasks and resources. Using this approximation, we develop a model predictive controller (MPC) [7] which generates the first stage allocations, then resolves the problem based on the outcome information available for the second stage allocations. We compare the performance of the MPC controller with that of the optimal SDP algorithm and a faster suboptimal SDP algorithm using random problems. Our results show that the model predictive algorithm achieves on average over 99% of the performance of the optimal SDP algorithm, while computing allocations for nearly 1000 tasks in under 1 second.

The rest of this paper is organized as follows: Sec-

¹This research was supported by AFOSR under Grant number F49620-01-1-0348, and by DARPA Information Systems Office and AFRL/VACA under contract number F33615-01-C-3149.

²ECE Department, Boston University, Boston, MA 02215, e-mail: dac@bu.edu

³ALPHATECH, Inc., 50 Mall Rd., Burlington, MA 01803, e-mail: jwohletz@alphatech.com

tion 2 describes the mathematical formulation. Section 3 discusses the exact SDP algorithm, and a simpler approximate SDP algorithm. Section 4 describes the fast approximate SDP formulation and its solution, and MPC approach. Section 5 discusses the numerical experiments. Due to space limitations, the proofs are omitted, and can be found in [4].

2 Problem Formulation

Assume that there are N tasks, indexed by $i = 1, \dots, N$, and that there are of M non-renewable homogeneous resources which can be assigned to each task over two possible stages. Associated with each task is a value V_i which is obtained by completing the task. Use of a resource incurs a cost C . When a resource is assigned to task i in stage k , the event that the resource successfully completes the task has probability $p_i(k)$, and this event is independent of any other events generated by other resource assignments.

Let $x_i(1)$ denote the number of resources assigned to task i at stage 1. Under the independence assumptions, the probability that task i is not completed is given by:

$$P_S(i, 1) = (1 - p_i(1))^{x_i(1)} \quad (1)$$

At the completion of stage 1, the set of completed tasks will be observed. Let $\Omega = \{0, 1\}^N$ denote the set of possible values of this observation, where $\omega_i = 0$ denotes that task i was completed in stage 1, and $\omega_i = 1$ denotes the complementary event for task i . Given a vector $\underline{x}(1)$, eq. (1) induces a probability distribution $P(\underline{\omega}|\underline{x}(1))$ on the possible outcomes. The stage 2 allocations are strategies $\underline{x}(2, \underline{\omega})$ that depend on the specific observed outcome. We refer to these strategies as *recourse strategies*.

Given resource allocations $\underline{x}(1)$ and recourse strategies $\underline{x}(2, \underline{\omega})$, the probability that task i is not completed either in stage 1 or in stage 2 is given by

$$P_S(i, 2) = \sum_{\underline{\omega} \in \Omega} P(\underline{\omega}|\underline{x}(1)) I(\omega_i = 1) (1 - p_i(2))^{x_i(2, \underline{\omega})} \quad (2)$$

where $I(\cdot)$ is the indicator function, and

$$P(\underline{\omega}|\underline{x}(1)) = \prod_{\{i|\omega_i=0\}} [1 - (1 - p_i(1))^{x_i(1)}] \prod_{\{j|\omega_j=1\}} (1 - p_j(1))^{x_j(1)}$$

The stochastic control problem is to select resource allocations $\underline{x}(1)$ and recourse strategies $\underline{x}(2, \underline{\omega})$ that minimize the expected incomplete task value plus the expected cost of using resources:

$$E\left\{\sum_{i=1}^N V_i P_S(i, 2) + C \sum_{i=1}^N (x_i(1) + x_i(2, \underline{\omega}))\right\} \quad (3)$$

subject to the constraints

$$\sum_{i=1}^N (x_i(1) + x_i(2, \underline{\omega})) \leq M \quad \text{for all } \underline{\omega} \in \Omega \quad (4)$$

$$x_i(1), x_i(2, \omega) \in \{0, 1, \dots, M\} \quad \text{for all } \omega \in \Omega, i \quad (5)$$

The above problem is a two-stage stochastic feedback control problem with a discrete state space that grows exponentially in the number of tasks N , and a decision space which grows exponentially in the number of tasks. In the next section, we discuss the solution of this problem using stochastic dynamic programming.

3 Stochastic Dynamic Programming Solution

Consider the problem at the second stage, after the state $\underline{\omega}$ has been observed. Without loss of generality, assume that there are N_2 incomplete tasks in $\underline{\omega}$, renumbered from $j = 1, \dots, N_2$, and that there are M_2 resources remaining. The second stage problem can be expressed as follows:

$$\min_{\{x_j, j=1, \dots, N_2\}} \sum_{j=1}^{N_2} \{V_j(1 - p_j)^{x_j} + Cx_j\} \quad (6)$$

subject to

$$\sum_{j=1}^{N_2} x_j \leq M_2, \quad x_j \in \{0, 1, \dots, M_2\} \quad (7)$$

Define real-valued functions over nonnegative integer allocation variables $n \in \{0, \dots, M_2\}$ as

$$f_j(n) = V_j[(1 - p_j)^n] + Cn \quad (8)$$

and define $f_j(x)$, $x \in [0, M_2]$ as the linear interpolation of the function $f_j(n)$. Note that $f_j(x)$ is convex in $x \in [0, M_2]$, as it is the sum of two convex functions. Relaxing the second stage optimization problem to allow for real-valued allocations results in a monotropic optimization problem [11] of the form

$$\min_{x_j \in [0, M_2]} \sum_{j=1}^{N_2} f_j(x_j) \quad (9)$$

subject to

$$\sum_{j=1}^{N_2} x_j \leq M_2, \quad x_j \geq 0 \quad (10)$$

The piecewise linear, convex nature of $f_j(x)$, together with the fact that M_2 is an integer and all the points of nondifferentiability of $f_j(x)$ correspond to integer x , guarantees that the solution to eq. (10) is an integer [11]. Furthermore, the separable convex nature

of the objective function and the single additive constraint leads to simple computations of subgradients, and guarantees the existence of a scalar Lagrange multiplier which satisfies the Karush-Kuhn-Tucker conditions [11]. The result is a fast algorithm for determining the optimal recourse allocations $x_2(i, \omega)$ and the corresponding optimal value $J_2(\omega, M_2)$, which is equivalent to an incremental line search for the optimal Lagrange multiplier value. The key structural result is stated below:

Lemma 1 Consider the second stage allocation problem defined by (9-10). There exists a nonnegative value λ^* such that the optimal resource allocation $\{x_j^*\}$ is given by

$$x_j^* \in \operatorname{argmin}_{x_j \in [0, M_2]} [f_j(x_j) + \lambda^* x_j] \quad (11)$$

Furthermore, λ^* can be chosen as the negative of one of the slopes of the piecewise linear functions $f_j(x)$.

The optimal solution has an *incremental optimality property*: The optimal solution for a given value of M_2 is part of an optimal solution for any value $M > M_2$. This leads to efficient algorithms for solving the second stage allocation problem, of complexity $O(N + M \ln(N + M))$ [5]. These algorithms are used for each possible first stage outcome ω and remaining resource level M_2 to compute the optimal cost-to-go $J_2^*(\omega, M_2)$. The optimal cost-to-go has the following properties:

Lemma 2 The optimal cost-to-go function $J_2^*(\omega, M_2)$ has the following properties:

1. $J_2^*(\omega, M_2)$ is a convex, piecewise linear, nonincreasing function of M_2 with breakpoints only at integer values of M_2 .
2. Consider two distinct outcomes $\omega^{(1)}, \omega^{(2)}$. If $\omega_i^{(1)} \leq \omega_i^{(2)}$ for $i = 1, \dots, N$, then $J_2^*(\omega^{(1)}, M_2) \leq J_2^*(\omega^{(2)}, M_2)$.

Consider now the first stage problem. The solution of eqs. (3-5) satisfies the stochastic dynamic programming recursion

$$J^* = \min_{\underline{x}(1) \in \{0, 1, \dots, M\}^N} \sum_{\omega} P(\omega | \underline{x}(1)) J_2^*(\omega, M - \sum_{i=1}^N x_i(1)) + C \sum_{i=1}^N x_i(1) \quad (12)$$

subject to the constraint

$$\sum_{i=1}^N x_i(1) \leq M \quad (13)$$

Unfortunately, the above optimization problem is a non-separable integer programming problem, and the objective function has 2^N terms in the summation. Exact solution of this problem is a difficult combinatorial problem. However, the presence of the single constraint (13) suggests the use of an incremental optimization approach similar to that used for the second stage problem, based on an incremental optimization approach, as follows: Define the notation \underline{x}_i^+ to denote the vector $(x_1 \dots x_{i-1} \ x_i + 1 \ x_{i+1} \dots x_N)^T$. Let $J(\underline{x})$ be defined as

$$J(\underline{x}) = \sum_{\omega} P(\omega | \underline{x}) J_2^*(\omega, M - \sum_{i=1}^N x_i) + C \sum_{i=1}^N x_i(1) \quad (14)$$

The incremental optimization algorithm can be described as:

1. Initialize $x_i = 0, i = 1, \dots, N$.
2. For each i , compute $MR_i(\underline{x}) = J(\underline{x}) - J(\underline{x}_i^+)$.
3. Select i^* for which $MR_{i^*}(\underline{x}) \geq MR_i(\underline{x})$ for all $i \neq i^*$.
4. If $MR_{i^*} > 0$ and $\sum_{i=1}^N x_i < M$, set $x_{i^*} = x_{i^*} + 1$; otherwise, stop.
5. Repeat steps 2-4 until algorithm stops.

Note that the solution to eqs. (12,13) is not guaranteed to have the incremental optimality property. Thus, the above algorithm is only an approximate algorithm, although our experimental results indicate its performance is indistinguishable from that of an enumerative search. Note also that computation of $MR_i(\underline{x})$ still requires summation over 2^N terms, an exponential complexity in the number of tasks. In the next section, we describe an alternative suboptimal approach, based on using Model Predictive Control [7] with an approximate optimization model, which can generate solutions in complexity $O((N + M) \ln N)$.

4 Model Predictive Control

In order to avoid the exponential growth in complexity as the number of tasks and resources grow, we propose an alternative algorithm based on model predictive control (MPC). The key to this approach is an aggregate model of the optimization problem represented by equations (3-5). This aggregate model is based on replacing the 2^N constraints in eq.(4) by one average resource utilization constraint. This "soft" constraint requires that the average number of resources across all sample paths cannot exceed the available number of resources. This approach is similar to the approach

used in [3, 13] for other dynamic resource allocation problems. Mathematically, the new constraint is:

$$\sum_{i=1}^N \sum_{\omega \in \Omega} P(\omega | \underline{x}(1)) (x_i(1) + x_i(2, \omega)) \leq M \quad (15)$$

The optimization problem used in the MPC approach is to minimize eq. (3) subject to constraints in eqs. (15,5). The solution determines $\underline{x}(1)$ as well as strategies for future allocations. Only the first stage allocations are implemented; subsequently, based on the observed outcome ω , the second stage allocations are determined using the approach discussed previously. Note that every strategy which was feasible for the constraints in eq.(4) satisfies the new constraint in eq.(15). Thus, the relaxed problem overestimates the expected performance that can be achieved with the available resources in the second stage. The key to the model predictive control approach is the efficient solution, over closed-loop strategies, of the model problem in eqs.(3,15,5). As a preliminary step, we expand the set of admissible strategies to include mixed strategies. That is, we introduce an additional random variable θ , independent of other random variables, with distribution $P(\theta)$ so that allowable decision strategies are of the form $\underline{x}(1, \theta), \underline{x}(2, \omega, \theta)$. Since the number of pure strategies is finite, the use of mixed strategies allows for the full utilization of the available resources in eq.(15). Note that, in the original stochastic dynamic programming, the use of mixed strategies does not change the optimal cost, as there exists an optimal solution which uses only pure strategies. However, the relaxed problem in eqs.(3,15,5) will typically have a better cost when mixed strategies are allowed, due to the knapsack nature of the integer allocation problem: mixed strategies will allow full utilization of the available resources in the constraint of eqn.(15).

Let (J^k, R^k) denote the expected performance and resource utilization of pure strategy k ; mixed strategies allow us to achieve any performance and resource utilization (J, R) in the convex hull of these expected performance-resource pairs. We define *local strategies* as follows:

Definition 1 A local strategy consists of a pure strategy $(\underline{x}(1), \underline{x}(2, \omega))$ with the property that $x_i(2, \omega) = x_i(2, \omega_i)$.

Thus, local strategies generate recourse allocations for individual tasks based on the observed state of that task only. In contrast, general strategies use recourse allocations that depend on the combined states of all the tasks ω .

Theorem 1 Consider the optimization problem in eqs. (3,15,5). Given any pure strategy, there is a

mixed strategy using only local strategies that achieves the same expected performance and the same expected resource use.

The result is based on the property that the objectives and the averaged constraints can be decomposed additively over tasks. This leads to an explicit construction of the mixed local strategies which have equivalent expected performance and expected resource use as a given pure strategy.

Let k denote an index over all local strategies, and let (J^k, R^k) denote the expected performance and resource utilization of strategy k . The optimal mixed strategy is the solution of the linear program

$$\min_{\theta^k} \sum_k J^k \theta^k \quad (16)$$

subject to

$$\sum_k \theta^k R^k \leq M; \quad \sum_k \theta^k = 1, \quad 0 \leq \theta^k \leq 1$$

This linear program is over mixtures of all local strategies, which is a large number. The next results provide a better characterization of the optimal strategy.

Lemma 3 There is an optimal mixed strategy which is a mixture of at most two local strategies.

Let (J_i^k, R_i^k) denote the expected performance and resource allocation for task i under local strategy k . Then,

$$J_i^k = \sum_{\omega_i} P(\omega_i | x_i(1)) \{ V_i I(\omega_i = 1) (1 - p_i(2))^{x_i(2, \omega_i)} + C x_i(2, \omega_i) \} + C x_i(1) \quad (17)$$

Define the functions $F_i(T_i)$ as the solution of the following single task resource allocation problem:

$$F_i(T) = \min_{\theta^k} \sum_k J_i^k \theta^k \quad (18)$$

subject to

$$\sum_k \theta^k R_i^k \leq T_i; \quad \sum_k \theta^k = 1, \quad 0 \leq \theta^k \leq 1 \quad (19)$$

Lemma 4 $F_i(T_i)$ are piecewise linear, convex, non-increasing functions of T_i .

Lemma 5 The corners of the functions $F_i(T_i)$ correspond to solutions the following equation for nonnegative values of λ

$$\min_{x, y \in \{0, \dots, M\}} (1 - p_i(1))^x \{ V_i (1 - p_i(2))^y + (C + \lambda)y \} + (C + \lambda)x \quad (20)$$

Furthermore, the optimizing solutions $x^*(\lambda), y^*(\lambda)$ are monotone nonincreasing in λ .

Using the above properties, one can develop a fast algorithm for computing the function $F_i(T_i)$ in complexity $O(M \log M)$ for each i , as described in [4].

With the above notation, we can rewrite the MPC problem using mixed local strategies as the following hierarchical problem

$$\max_{T_i, i=1, \dots, N} \sum_i F_i(T_i) \quad (21)$$

subject to

$$\sum_i T_i \leq M, \quad T_i \geq 0, i = 1, \dots, N \quad (22)$$

This is another monotropic programming problem, of the type discussed earlier in the second stage of dynamic programming. The only difference is that the corner points of $F_i(T_i)$ do not occur at integer values of T_i . The optimal solution is obtained by the same algorithm: the negative of slopes of $F_i(T_i)$ segments are possible values of the Lagrange multiplier λ associated with transitions in resource allocations. The maximum number of possible transition values of λ is $2M$ per task, for a total less than or equal to $2MN$. Performing a line search over this value results in a polynomial time algorithm for exact solution of the Model Predictive Control problem; the solution will have the property that only one task (corresponding to a negative slope equal to the final value of λ will use a local mixed strategy. A faster algorithm that computes the slopes incrementally for each i , and keeps track only of the next slope for each task, can be shown to solve the problem in complexity $O((M + N) \log N)$.

Once the model predictive control solution is determined, the first stage allocations are assigned to each task. The only ambiguity occurs when the local mixed strategy for the final task is a mixture of two different first stage allocations; in this case, we allocate the smaller of the two first stage allocations to that task.

5 Experimental Results

In order to evaluate the effectiveness of the proposed MPC approach, we conducted several experiments with the following algorithms:

1. The exact SDP solution, obtained by enumerating the possible first stage allocations and finding a global minimum.
2. The Incremental DP (IA) algorithm discussed at the end of Section 3.
3. The MPC algorithm described in Section 4.

Tasks	Resources	IA Alg.	MPC Alg.	Worst MPC
7	7	100%	99.92%	98.6%
7	9	100%	99.82%	99.18%
7	11	100%	99.996%	99.86%
9	7	100%	99.91%	98.30%
9	9	100%	99.89%	99.48%
9	11	100%	99.82%	98.96%
11	7	100%	99.92%	99.53%
11	9	100%	99.93%	99.56%
11	11	100%	99.74%	99.19%

Table 1: Performance of the Model Predictive Control (MPC) and Incremental Algorithm (IA) as percent of value completed by DP

The first set of experiments consisted of random problems with 7 to 11 tasks, with task values selected randomly in the range of 1-10, and task success probabilities selected randomly in the interval [0.7, 0.9]. The number of resources for each number of task varied from 7 to 11 resources. For each number of tasks, we generated 100 random problems, and obtained the optimal solution (in terms of value achieved) by SDP, IA and MPC algorithms. The statistics in the experiment report the percentage of the value achieved by the optimal SDP algorithm averaged over the 100 problems. We also computed the worst case percentage difference in performance between the MPC algorithm and the SDP algorithm. The results are summarized in Table I. The results indicate that the performance of IA was optimal for all random problems generated. The results also show that the MPC algorithm yields near-optimal performance: The worst case performance across 900 problems tested was within 2% of the optimal SDP performance, and the average performance was within 0.3% of the optimal SDP performance.

The second set of experiments used problems with 16 and 20 tasks, and with a varying number of resources from 12 to 20. For these problems, computing the exact dynamic programming solution using enumerative techniques was prohibitively long. As a reference point, it required 3 days on a LINUX Pentium 1.7 GHz workstation to solve 100 instances of the 11 task problem. We compared results only for IA and MPC algorithms. The statistics reported are the percentage of the value achieved by the IA algorithm. The results are summarized in Table II. The results in Table II confirm the near optimal behavior of the Model Predictive Control algorithm. The average performance is within 0.2% of the performance of the IA algorithm, and the worst case performance is within 1% of the performance of the IA algorithm. The experiments confirm that the MPC algorithm's bias to commit more resources in the first stage has a nearly negligible impact in overall task performance.

Tasks	Resources	Ave. MPC	Worst MPC
16	12	99.81%	99.22%
16	16	99.82%	99.33%
16	20	99.92%	99.67%
20	12	99.85%	99.46%
20	16	99.85%	99.52%
20	20	99.88%	99.37%

Table 2: Performance of the Model Predictive Control (MPC) algorithm as percent of value completed by Incremental Algorithm (IA) for different numbers of tasks.

The IA algorithms required over 13 minutes to solve a single instance of a 20 task, 20 resource problem on a Pentium 1.4 GHz workstation running Linux. In contrast, the MPC algorithm solved 100 instances of 1000 task, 1000 resource problems in a total of 3.5 seconds. This suggests that the MPC algorithm is well suited to applications where information about available tasks and values becomes available in real time, and must be converted into resource allocation decisions quickly.

6 Conclusion

The problem of allocation of unreliable resources to tasks over multiple stages arises in many important applications. In this paper, we have developed a stochastic dynamic programming formulation for this problem, which captures the opportunity for observing task completion events and using recourse strategies. However, exact solution of this problem using Stochastic Dynamic Programming is computationally intensive because both the state space and the admissible action space grow exponentially with the number of tasks. As an efficient alternative, we developed a Model Predictive Control algorithm that is based on solving a relaxed stochastic dynamic programming problem. We established that the relaxed problem can be solved very fast, in time nearly linear with the number of tasks. Furthermore, the resulting algorithm exhibits near-optimal performance across a range of random test problems.

There are several important directions for extension of this work which have been pursued in [4]. The first of these is extension of the results to multiple resource classes and multiple stages. The main theorem in this paper, the representation of the optimal relaxed strategies in terms of local strategies, extends in a straightforward manner to these cases. Another interesting extension is to consider tasks that require multiple assignment of simultaneous resources to complete. Extensions of our techniques to this problem, and problems where tasks have precedence constraints, are currently

under investigation.

References

- [1] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, Belmont, MA, 2000.
- [2] D. A. Castañón, *Advanced Weapon Target Assignment Algorithms*, ALPHATECH report TR 428, Burlington, MA, 1989.
- [3] D. A. Castañón "Approximate Dynamic Programming for Sensor Management," *Proc. 36th IEEE Conference on Decision and Control*, San Diego, CA, December 1997.
- [4] D. A. Castañón and J. M. Wohletz, "Model Predictive Control for Dynamic Resource Allocation," submitted to *IEEE Transactions on Automatic Control*.
- [5] A. R. Eckler and S. A. Burr, *Mathematical Models of Target Coverage and Missile Allocation*, Military Operations Research Society, Alexandria, VA 1972.
- [6] P. Hosein, *A Class of Nonlinear Resource Allocation Problems*, MIT Ph. D. Thesis, Electrical and Computer Engineering, 1989.
- [7] D. Q. Mayne, J. B. Rawlings, C. V. Rao and P. O. M. Skokaert, "Constrained Model Predictive Control: Stability and Optimality," *Automatica*, V. 36, 2000.
- [8] R. A. Murphey, "An Approximate Algorithm for a Weapon Target Assignment Program," in *Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems*, P. Pardalos (ed.), Kluwer Academic, Dordrecht, 1999.
- [9] R. A. Murphey, "Target-Based Weapon Target Assignment Problems," in *Nonlinear Assignment Problems: Algorithms and Applications*, P. M. Pardalos and L. S. Pitsoulis, eds., Kluwer Academic, 2000.
- [10] R.T. Rockafellar and R.J.-B. Wets, "Stochastic convex programming: Kuhn-Tucker conditions," *J. Math. Econom.*, V. 2, 1975.
- [11] R. T. Rockafellar, *Network Flows and Monotropic Programming*, Wiley, New York 1984.
- [12] L. Stone and A. Washburn, eds., "Special Issue on Search Theory," *Naval Research Logistics*, Vol. 38, No. 4, 1991.
- [13] K. A. Yost and A. R. Washburn, "The LP/POMDP Marriage: Optimization with Imperfect Information," *Naval Research Logistics*, Vol 47, No. 8, 607-619, 2000.